
Università degli Studi di Milano
Master Degree in Computer Science

Information Management course

Teacher: Alberto Ceselli

Lecture 19: 18/12/2012

Data Mining: Concepts and Techniques


(3rd ed.)

— Chapter 10 —

Jiawei Han, Micheline Kamber, and Jian Pei
University of Illinois at Urbana-Champaign &
Simon Fraser University

©2011 Han, Kamber & Pei. All rights reserved.

Cluster Analysis: Concepts and Methods

- Cluster Analysis: Basic Concepts 
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

What is Cluster Analysis?

- Cluster: A collection of data objects
 - similar (or related) to one another within the same group
 - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation*, ...)
 - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- **Unsupervised learning**: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
 - As a **stand-alone tool** to get insight into data distribution
 - As a **preprocessing step** for other algorithms

Applications of Cluster Analysis

- Data reduction
 - Summarization: Preprocessing for regression, PCA, classification, and association analysis
 - Compression: Image processing: vector quantization
- Hypothesis generation and testing
- Prediction based on groups
 - Cluster & find characteristics/patterns for each group
- Finding K-nearest Neighbors
 - Localizing search to one or a small number of clusters
- Outlier detection: Outliers are often viewed as those “far away” from any cluster

Clustering: Application Examples

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market research

Basic Steps to Develop a Clustering Task

- Feature selection
 - Select info concerning the task of interest
 - Minimal information redundancy
- Proximity measure
 - Similarity of two feature vectors
- Clustering criterion
 - Expressed via a cost function or some rules
- Clustering algorithms
 - Choice of algorithms
- Validation of the results
 - Validation test (also, *clustering tendency* test)
- Interpretation of the results
 - Integration with applications

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
 - high intra-class similarity: **cohesive** within clusters
 - low inter-class similarity: **distinctive** between clusters
- The quality of a clustering method depends on
 - the similarity measure used by the method
 - its implementation (optimality guarantees + computational effectiveness), and
 - Its ability to discover some or all of the hidden patterns (practical behavior)

Measure the Quality of Clustering

- Dissimilarity/Similarity metric
 - Similarity is expressed in terms of a pairwise distance function, typically metric: $d(i, j)$
 - The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables
 - Weights should be associated with different variables based on applications and data semantics
- Quality of clustering:
 - There is usually a separate global quality function that measures the “goodness” of a cluster.
 - It is hard to define “similar enough” or “good enough” (need to stick to the application!)
 - The answer is typically highly subjective (i.e. don't blame the *algorithm* for *modeling* errors)

Considerations for Cluster Analysis

- Partitioning criteria
 - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
 - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- Similarity measure
 - Distance-based (e.g., Euclidian, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- Clustering space
 - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)


Requirements and Challenges

- Scalability
 - Clustering all the data instead of only on samples
- Ability to deal with different types of attributes
 - Numerical, binary, categorical, ordinal, linked, and mixture of these
- Constraint-based clustering
 - User may give inputs on constraints
 - Use domain knowledge to determine input parameters
- Interpretability and usability
- Others
 - Discovery of clusters with arbitrary shape
 - Ability to deal with noisy data
 - Incremental clustering and insensitivity to input order
 - High dimensionality

Major Clustering Approaches

- Partitioning approach:
 - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
 - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
 - Create a hierarchical decomposition of the set of data (or objects) using some criterion
 - Agglomerative (bottom-up) or divisive (top-down)
- Density-based approach:
 - Based on connectivity and density functions (keep growing as points are still in the neighborhood of cluster elements)
 - Find arbitrarily shaped clusters
- Grid-based approach:
 - Quantize object space in a grid structure
 - build a multiple-level granularity structure

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods 
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Partitioning Algorithms: Basic Concept

Partitioning method: Partitioning a database \mathbf{D} of n objects into a set of k clusters, such that the sum of squared distances with respect to a cluster representative c_i is minimized (e.g. the centroid or medoid of cluster C_i)

$$E = \sum_{i=1}^k \sum_{p \in C_i} (d(p, c_i))^2$$

Given k , find a partition of k clusters that optimizes the chosen partitioning criterion

- Global optimal: exhaustively enumerate all partitions
- Heuristic methods: *k-means* and *k-medoids* algorithms
- *k-means* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
- *k-medoids* or k-medians or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

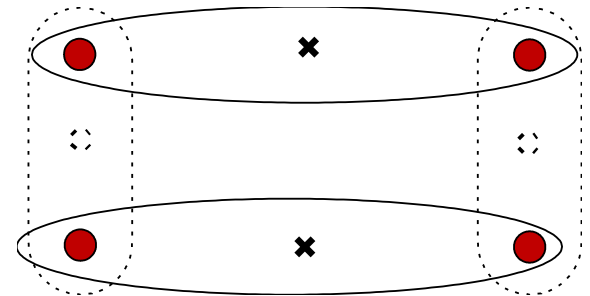
- Given k , the *k-means* algorithm is implemented in four steps:
 - Partition objects into k nonempty subsets
 - Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
 - Assign each object to the cluster with the nearest seed point
 - Go back to Step 2, stop when the assignment does not change

Comments on the *K-Means* Method

- Strength: *Efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t \ll n$.
 - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimal*
- Weakness
 - Applicable only to objects in a continuous n -dimensional space
 - Using the k -modes method for categorical data
 - In comparison, k -medoids can be applied to a wide range of data
 - Need to specify k , the *number* of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009))
 - Sensitive to noisy data and *outliers*
 - Not suitable to discover clusters with *non-convex shapes*

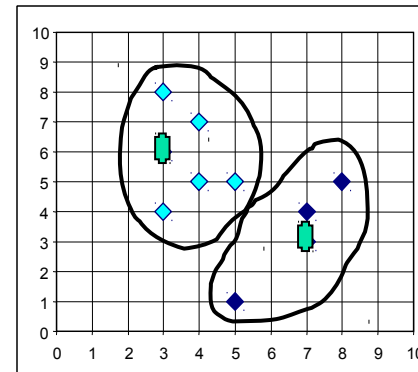
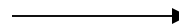
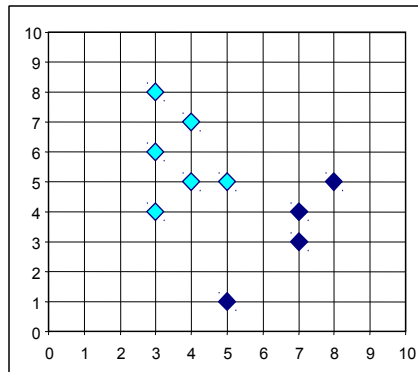
Variations of the *K-Means* Method

- Most of the variants of the *k-means* which differ in
 - Selection of the initial *k* means
 - Dissimilarity calculations
 - Strategies to calculate cluster means
- Handling categorical data: *k-modes*
 - Replacing means of clusters with modes
 - Using new dissimilarity measures to deal with categorical objects
 - Using a frequency-based method to update modes of clusters
 - A mixture of categorical and numerical data: *k-prototype* method

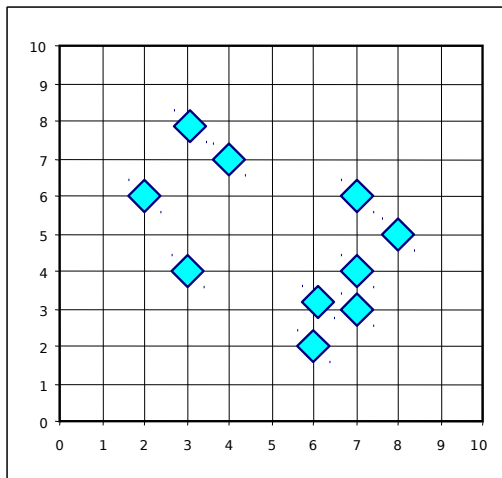


What Is the Problem of the K-Means Method?

- The k-means algorithm is sensitive to outliers !
 - Since an object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster

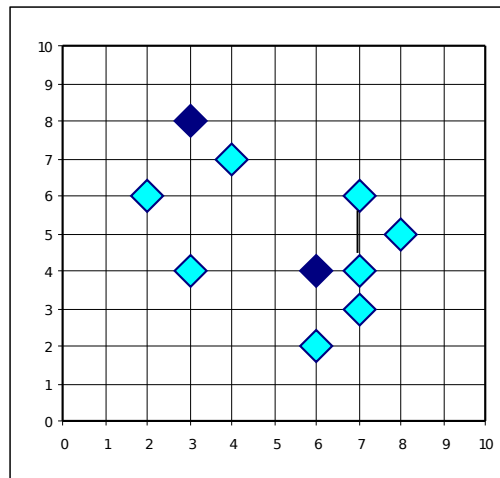


PAM: A Typical K-Medoids Algorithm



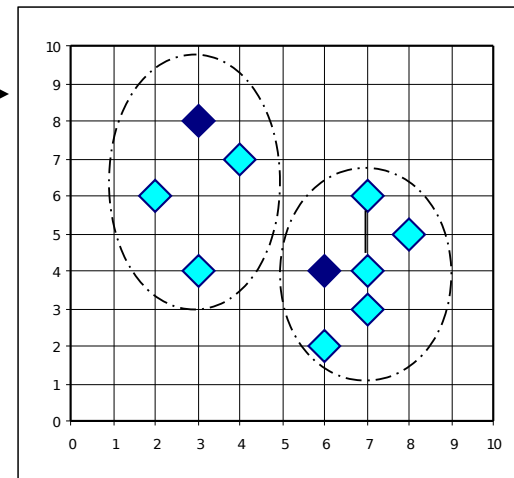
$K=2$

Arbitrarily choose k objects as initial medoids



Total Cost = 26

Assign each remaining object to nearest medoid

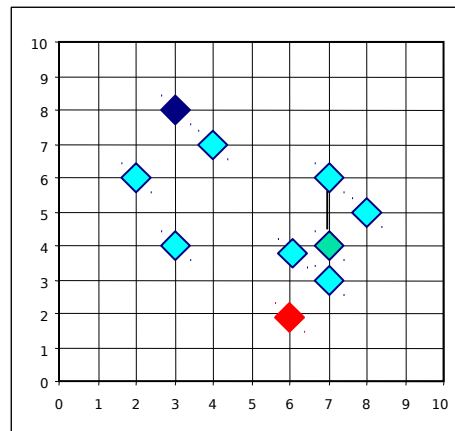


Total Cost = 20

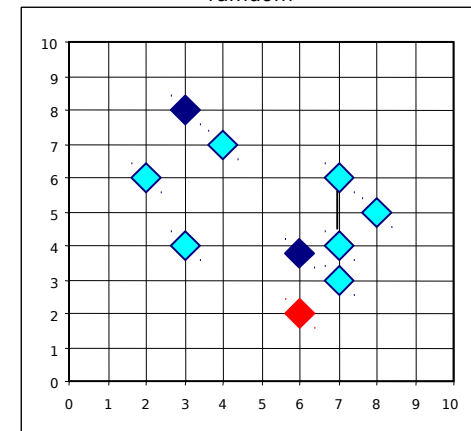
Randomly select a nonmedoid object, O_{random}

**Do loop
Until no
change**

Swapping O and O_{random}
If quality is improved.




Compute total cost of swapping



The K-Medoid Clustering Method

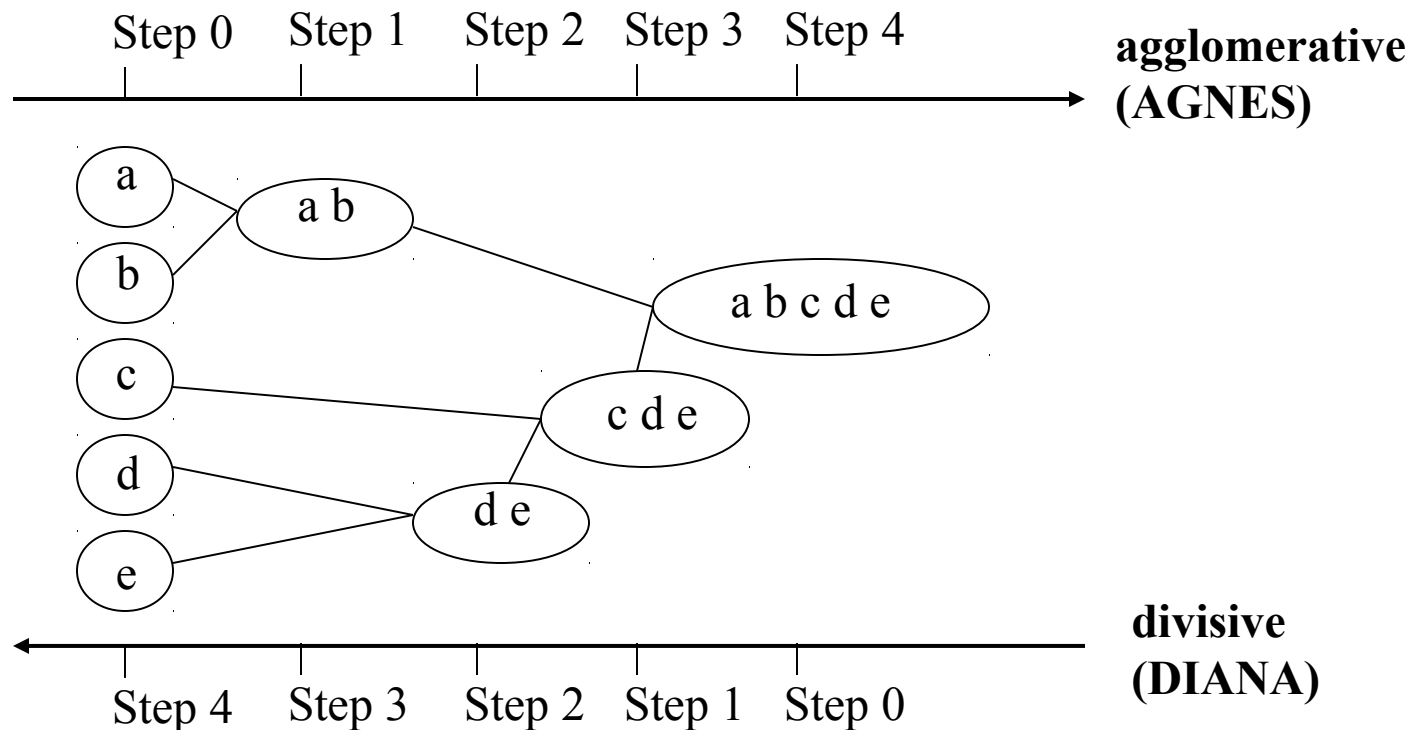
- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
 - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
 - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
 - *PAM* works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
- Efficiency improvement on PAM
 - *CLARA* (Kaufmann & Rousseeuw, 1990): PAM on samples
 - *CLARANS* (Ng & Han, 1994): Randomized re-sampling

Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods 
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

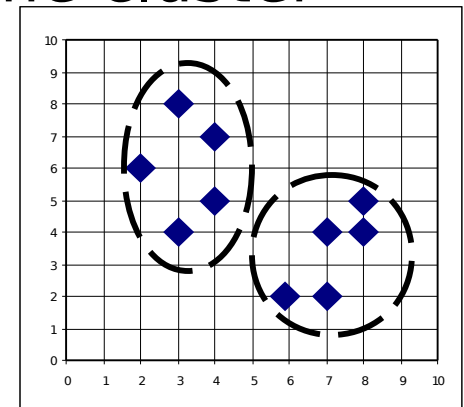
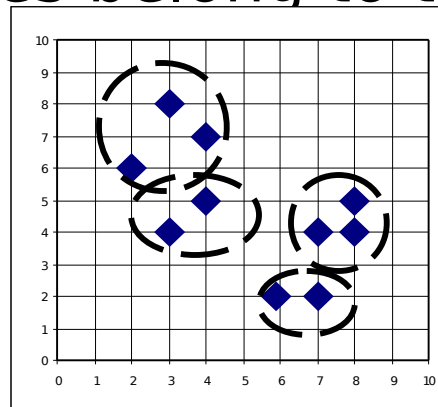
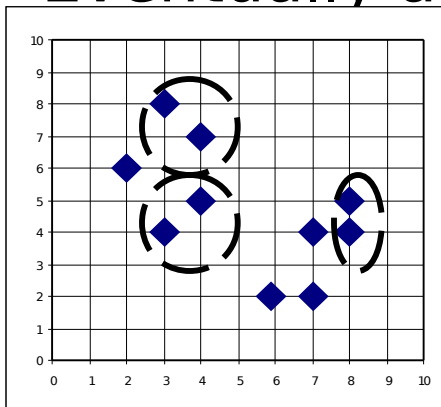
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters k as an input, but needs a termination condition



AGNES (AGglomerative NESTing)

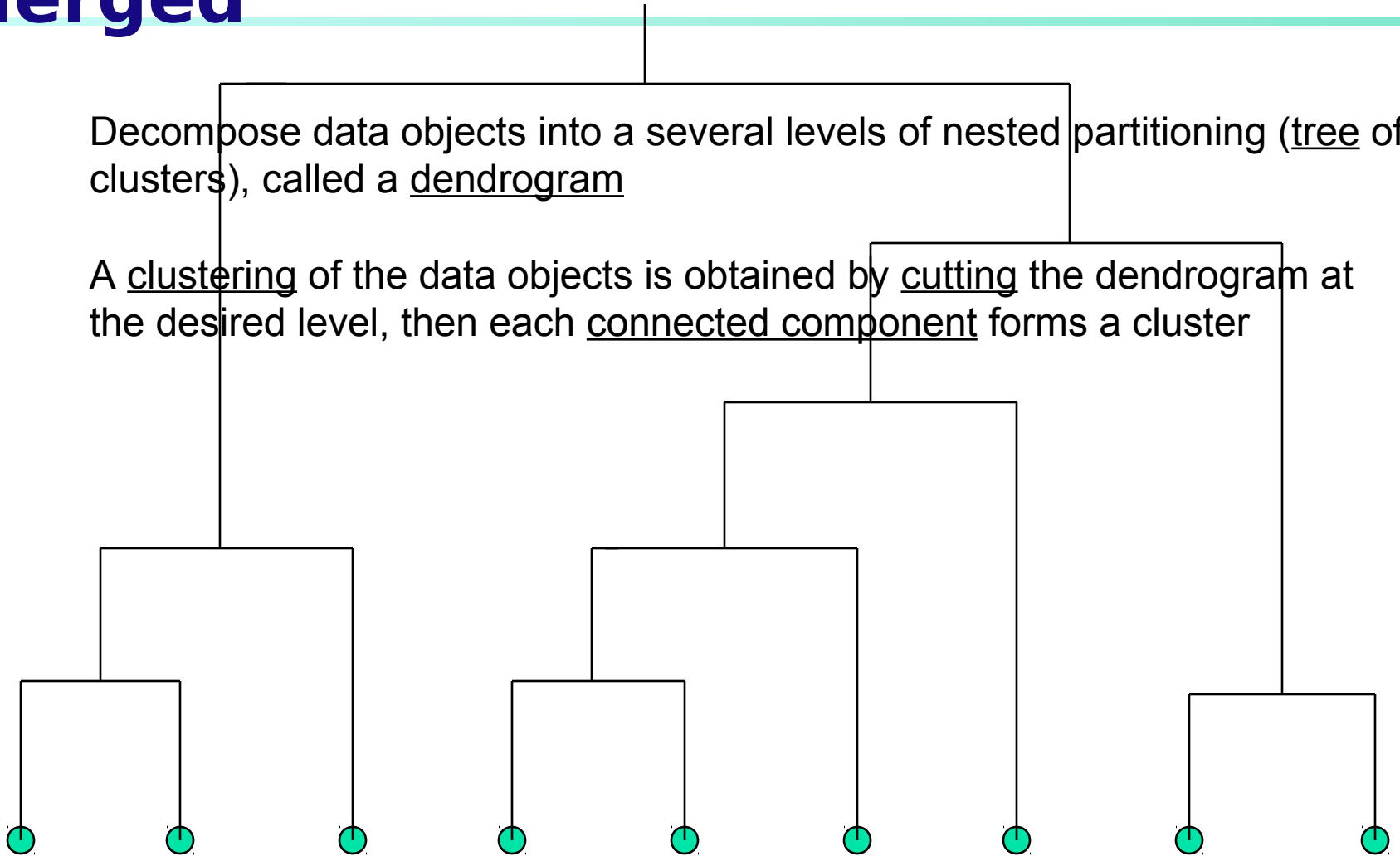
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the **single-link** method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



Dendrogram: Shows How Clusters are Merged

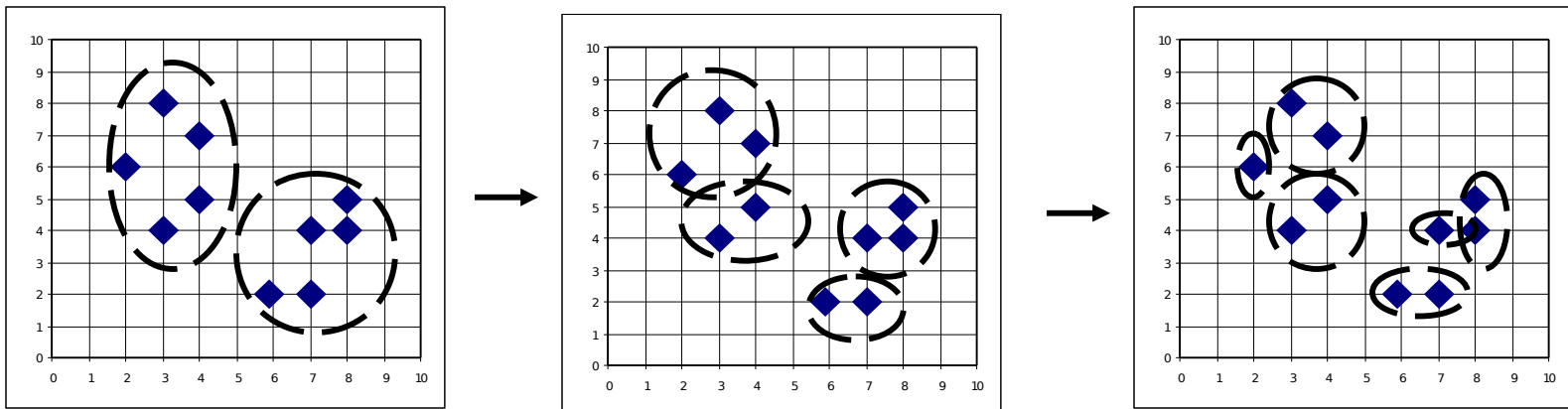
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster

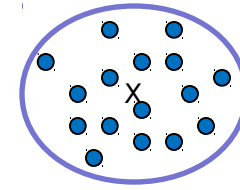
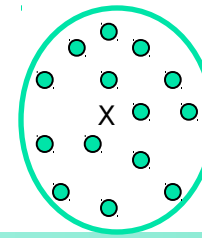


DIANA (Dlvisive ANAlysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own

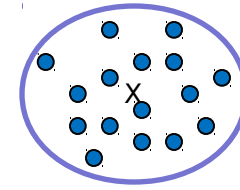
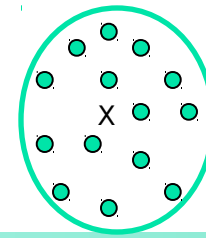


Distance between Clusters



- **Single link:** smallest distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \min_{p \in K_i, q \in K_j} d(p, q)$
- **Complete link:** largest distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \max_{p \in K_i, q \in K_j} d(p, q)$
- **Average:** avg distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \text{sum}_{p \in K_i, q \in K_j} d(p, q) / (|K_i||K_j|)$
- **Centroid:** distance between the centroids of two clusters, e.g., $p = \text{mean}(K_i)$, $q = \text{mean}(K_j)$, $\text{dist}(K_i, K_j) = d(p, q)$
- **Medoid:** distance between the medoids of two clusters, i.e., $p = \text{median}(K_i)$, $q = \text{median}(K_j)$, $\text{dist}(K_i, K_j) = d(p, q)$

Distance between Clusters



- Algorithms using minimum distance are also called nearest-neighbor clustering algorithms
 - they build minimum spanning trees
 - if clustering is terminated when the minimum inter-cluster distance exceeds a given threshold they are called single-linkage
- Algorithms using maximum distance are also called farthest-neighbor clustering algorithms
 - If clustering is terminated when the maximum inter-cluster distance between nearest clusters exceeds a given threshold they are called complete-linkage

Centroid, Radius and Diameter of a Cluster (for numerical data sets)

- Centroid: the “middle” of a cluster

$$C_m = \frac{\sum_{i=1}^N (t_{ip})}{N}$$

- Radius: square root of average distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^N (t_{ip} - c_m)^2}{N}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^N \sum_{i=1}^N (t_{ip} - t_{iq})^2}{N(N-1)}}$$

Extensions to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
 - Can never undo what was done previously
 - Do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
- Integration of hierarchical & distance-based clustering
 - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
 - CHAMELEON (1999): hierarchical clustering using dynamic modeling

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

- Zhang, Ramakrishnan & Livny, SIGMOD'96
- Clustering Feature (CF): $\langle n, LS, SS \rangle$
 - n : number of points, LS : their sum, SS : their sum of squares
- Easy to compute centroid, radius and diameter from CF
- CFs are additive
- Incrementally construct a CF tree, a hierarchical data structure for multiphase clustering
 - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve its inherent clustering structure)
 - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness*: handles only numeric data, and sensitive to the order of the data record

Clustering Feature Vector in BIRCH

Clustering Feature (CF): $CF = (N, LS, SS)$

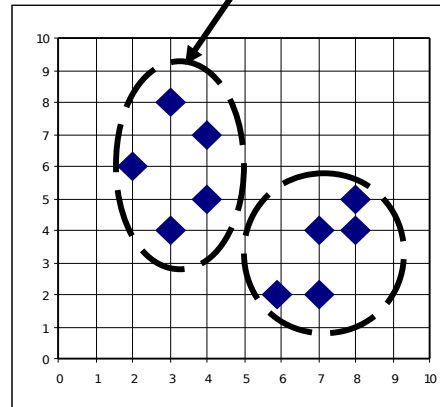
N : Number of data points

LS : linear sum of N points:

$$\sum_{i=1}^N X_i$$

SS : square sum of N points

$$\sum_{i=1}^N X_i^2$$



$CF = (5, (16,30),(54,190))$

(3,4)

(2,6)

(4,5)

(4,7)

(3,8)

CF-Tree in BIRCH

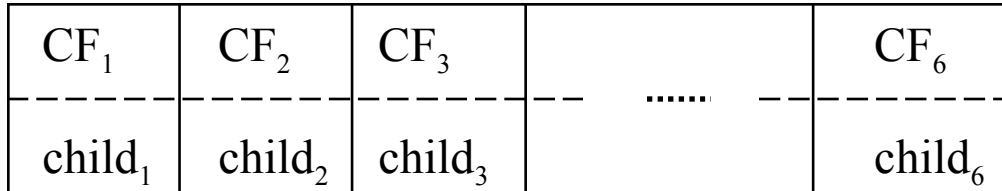
- Clustering feature:
 - Summary of the statistics for a given subcluster: the 0-th, 1st, and 2nd moments of the subcluster from the statistical point of view
 - Registers crucial measurements for computing cluster and utilizes storage efficiently
- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
 - A nonleaf node in a tree has descendants or “children”
 - The nonleaf nodes store sums of the CFs of their children
- A CF tree has two parameters
 - Branching factor: max # of children
 - Threshold: max diameter of sub-clusters stored at the leaf nodes

The CF Tree Structure

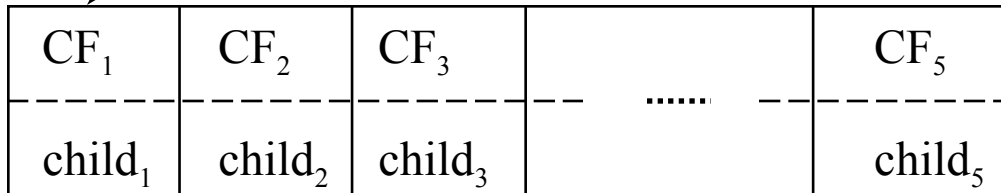
Root

$B = 7$

$L = 6$

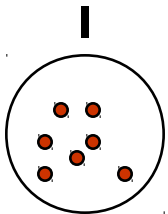
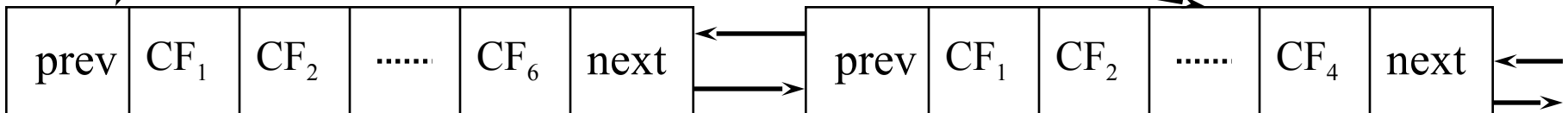


Non-leaf node



Leaf node

Leaf node



The Birch Algorithm

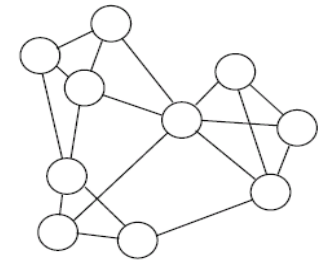
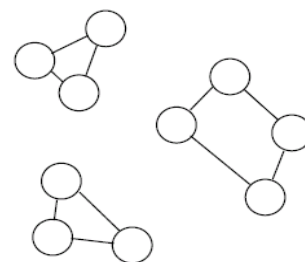
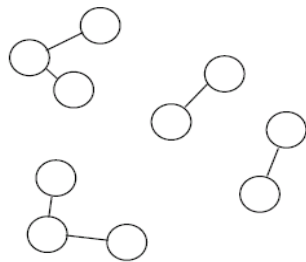
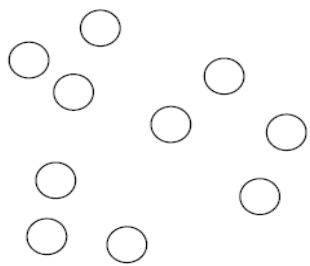
- Cluster Diameter $\sqrt{\frac{1}{n(n-1)} \sum (x_i - x_j)^2}$
- For each point in the input
 - Find closest leaf entry
 - Add point to leaf entry and update CF
 - If entry diameter $>$ max_diameter, then split leaf, and possibly parents
- Algorithm is $O(n)$
- Concerns
 - Sensitive to insertion order of data points
 - Since we fix the size of leaf nodes, so clusters may not be so natural
 - Clusters tend to be spherical given the radius and diameter measures

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling (1999)

- CHAMELEON: G. Karypis, E. H. Han, and V. Kumar, 1999
- Measures the similarity based on a dynamic model
 - Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters
- Graph-based, and a two-phase algorithm
 1. Use a graph-partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
 2. Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters

KNN Graphs & Interconnectivity

- k-nearest graphs from an original data in 2D:



(a) Original Data in 2D

(b) 1-nearest neighbor graph

(c) 2-nearest neighbor graph

(d) 3-nearest neighbor graph

- $EC_{\{C_i, C_j\}}$: The absolute inter-connectivity between C_i and C_j ; the sum of the weight of the edges that connect vertices in C_i to vertices in C_j
- Internal inter-connectivity of a cluster C_i : the size of its min-cut bisector EC_{C_i} (i.e., the weighted sum of edges that partition the graph into two roughly equal parts)
- Relative Inter-connectivity (RI):

$$RI(C_i, C_j) = \frac{|EC_{\{C_i, C_j\}}|}{\frac{|EC_{C_i}| + |EC_{C_j}|}{2}}$$

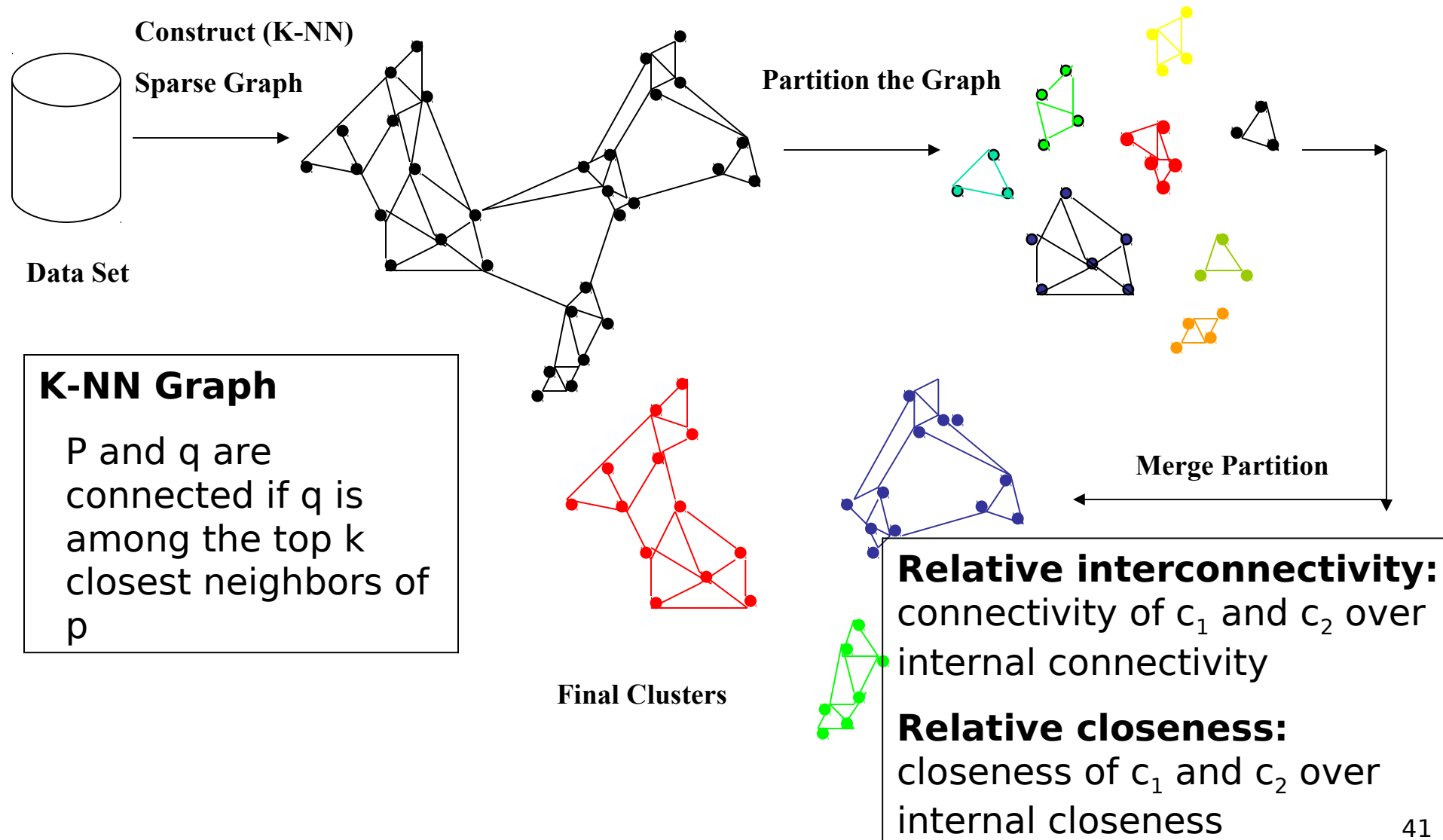
Relative Closeness & Merge of Sub-Clusters

- **Relative closeness** between a pair of clusters C_i and C_j : *the absolute closeness between C_i and C_j normalized w.r.t. the internal closeness of the two clusters C_i and C_j*

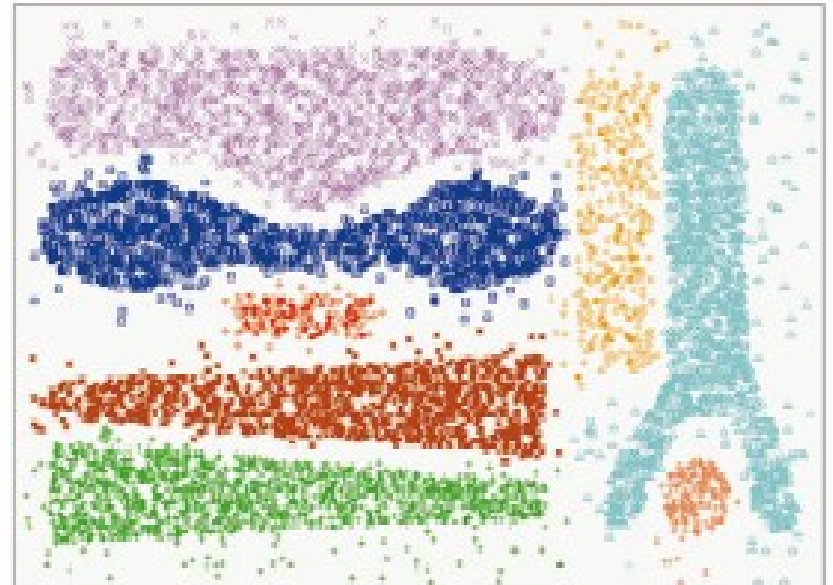
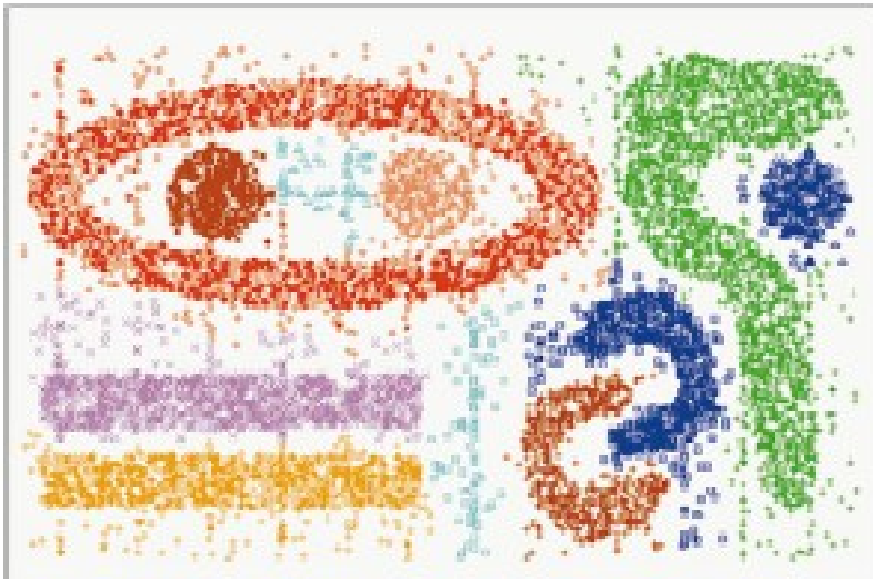
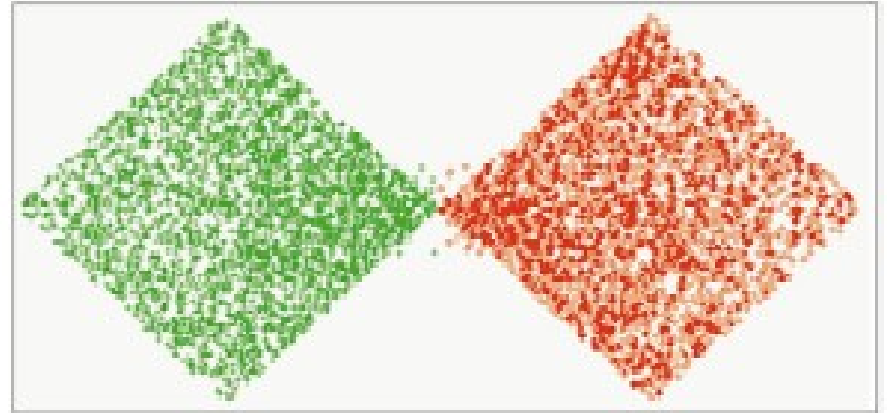
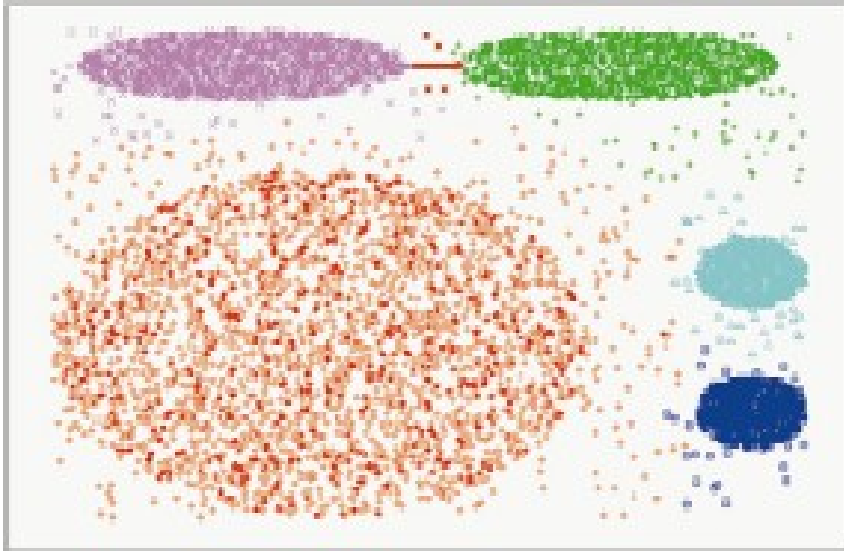
$$RC(C_i, C_j) = \frac{\bar{S}_{EC\{C_i, C_j\}}}{\frac{|C_i|}{|C_i|+|C_j|} \bar{S}_{EC C_i} + \frac{|C_j|}{|C_i|+|C_j|} \bar{S}_{EC C_j}}$$

- $\bar{S}_{EC C_i}$ and $\bar{S}_{EC C_j}$ are the average weights of the edges that belong in the min-cut bisector of clusters C_i and C_j , respectively, and $\bar{S}_{EC\{C_i, C_j\}}$ is the average weight of the edges that connect vertices in C_i to vertices in C_j
- **Merge Sub-Clusters:**
 - Merges only those pairs of clusters whose RI and RC are both above some user-specified thresholds
 - Merge those maximizing a function combining RI & RC

Overall Framework of CHAMELEON



CHAMELEON (Clustering Complex Objects)



Probabilistic Hierarchical Clustering

- Algorithmic hierarchical clustering
 - Nontrivial to choose a good distance measure
 - Hard to handle missing attribute values
 - No optimality guarantees: heuristic, local search
- Probabilistic hierarchical clustering
 - Use probabilistic models to measure distances between clusters
 - Generative model: Regard the set of data objects to be clustered as a sample of the underlying data generation mechanism to be analyzed
 - Easy to understand, same efficiency as algorithmic agglomerative clustering method, can handle partially observed data
- In practice, assume the generative models adopt common distributions functions, e.g., Gaussian distribution or Bernoulli distribution, governed by parameters

Generative Model

- Given a set of 1-D points $X = \{x_1, \dots, x_n\}$ for clustering analysis & assuming they are generated by a Gaussian distribution:

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- The probability that a point $x_i \in X$ is generated by the model

$$P(x_i | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

- The likelihood that X is generated by the model:

$$L(\mathcal{N}(\mu, \sigma^2) : X) = P(X | \mu, \sigma^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

- The task of learning the generative model: find the parameters μ and σ^2 such that

$$\mathcal{N}(\mu_0, \sigma_0^2) = \arg \max \{L(\mathcal{N}(\mu, \sigma^2) : X)\}$$

the maximum
likelihood

A Probabilistic Hierarchical Clustering Algorithm

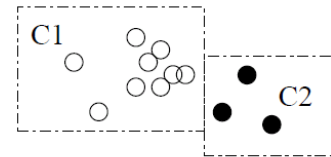
- For a set of objects partitioned into m clusters C_1, \dots, C_m , the quality can be measured by,

$$Q(\{C_1, \dots, C_m\}) = \prod_{i=1}^m P(C_i)$$

where $P()$ is the maximum likelihood

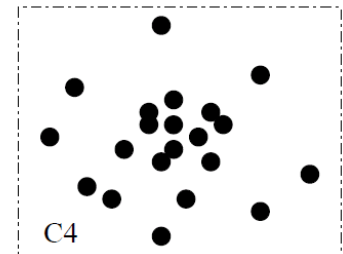
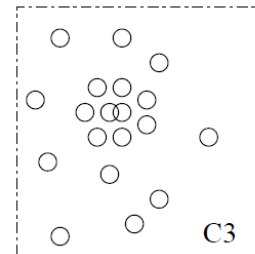
- If we merge two clusters C_{j_1} and C_{j_2} into a cluster $C_{j_1} \cup C_{j_2}$, then, the change in quality of the overall clustering is

$$\begin{aligned}
 & Q(\{C_1, \dots, C_m\} - \{C_{j_1}, C_{j_2}\} \cup \{C_{j_1} \cup C_{j_2}\}) - Q(\{C_1, \dots, C_m\}) \\
 = & \frac{\prod_{i=1}^m P(C_i) \cdot P(C_{j_1} \cup C_{j_2})}{P(C_{j_1})P(C_{j_2})} - \prod_{i=1}^m P(C_i) \\
 = & \prod_{i=1}^m P(C_i) \left(\frac{P(C_{j_1} \cup C_{j_2})}{P(C_{j_1})P(C_{j_2})} - 1 \right)
 \end{aligned}$$




- Distance between clusters C_1 and C_2 :

$$dist(C_i, C_j) = -\log \frac{P(C_1 \cup C_2)}{P(C_1)P(C_2)}$$



Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods 
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Density-Based Clustering Methods

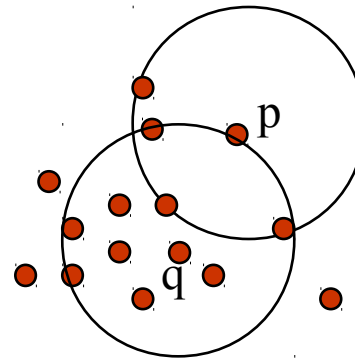
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
 - Discover clusters of arbitrary shape
 - Handle noise
 - One scan
 - Need density parameters as termination condition
- Several interesting studies:
 - DBSCAN: Ester, et al. (KDD'96)
 - OPTICS: Ankerst, et al (SIGMOD'99).
 - DENCLUE: Hinneburg & D. Keim (KDD'98)
 - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

Density-Based Clustering: Basic Concepts

- Two parameters:
 - *Eps*: Maximum radius of the neighbourhood
 - *MinPts*: Minimum number of points in an *Eps*-neighbourhood of that point
- $N_{Eps}(q)$: $\{p \text{ belongs to } D \mid \text{dist}(p,q) \leq Eps\}$
- **Directly density-reachable**: A point p is directly density-reachable from a point q w.r.t. *Eps*, *MinPts* if

- p belongs to $N_{Eps}(q)$
- core point condition:

$$|N_{Eps}(q)| \geq MinPts$$

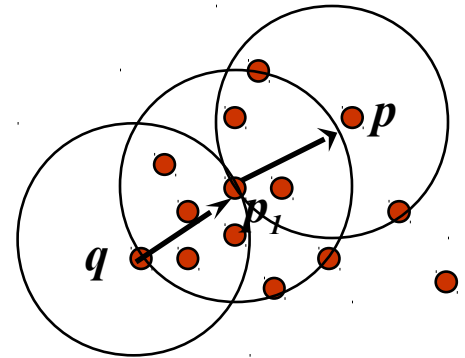


MinPts = 5

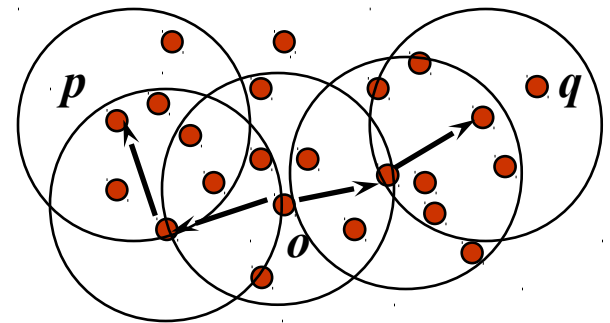
Eps = 1 cm

Density-Reachable and Density-Connected

- Density-reachable:
 - A point p is **density-reachable** from a point q w.r.t. Eps , $MinPts$ if there is a chain of points p_1, \dots, p_n , $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i

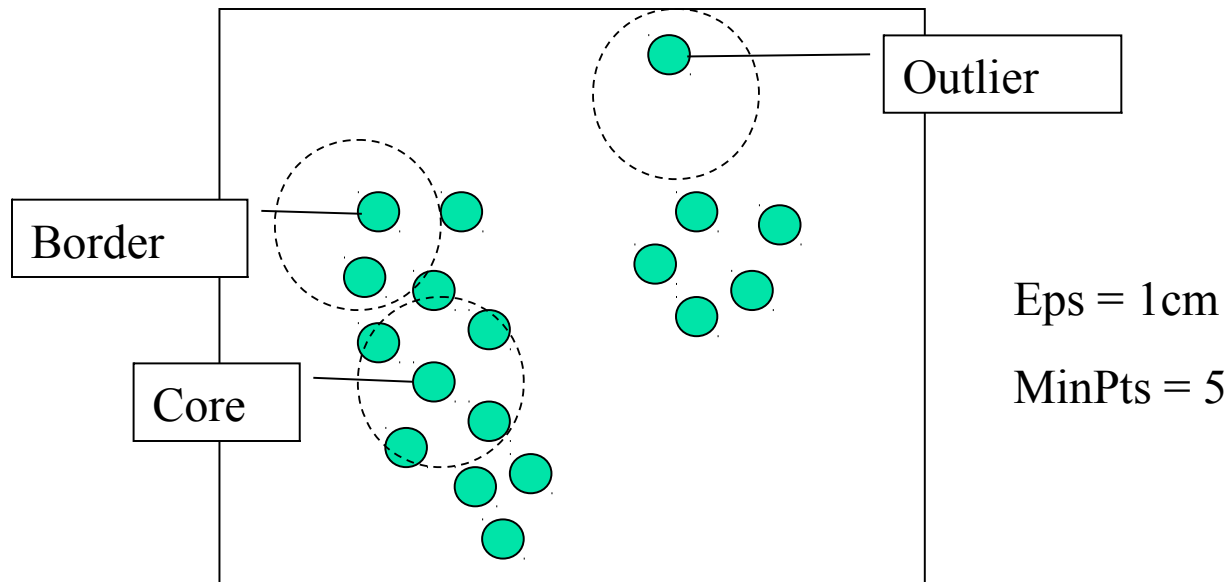


- Density-connected
 - A point p is **density-connected** to a point q w.r.t. Eps , $MinPts$ if there is a point o such that both, p and q are density-reachable from o w.r.t. Eps and $MinPts$



DBSCAN: Density-Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: The Algorithm

- Arbitrary select a point p
- Retrieve all points density-reachable from p w.r.t. Eps and $MinPts$
- If p is a core point, a cluster is formed
- If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database
- Continue the process until all of the points have been processed
- If a spatial index is used, the computational complexity of DBSCAN is $O(n \log n)$, where n is the number of database objects. Otherwise, the complexity is $O(n^2)$

DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

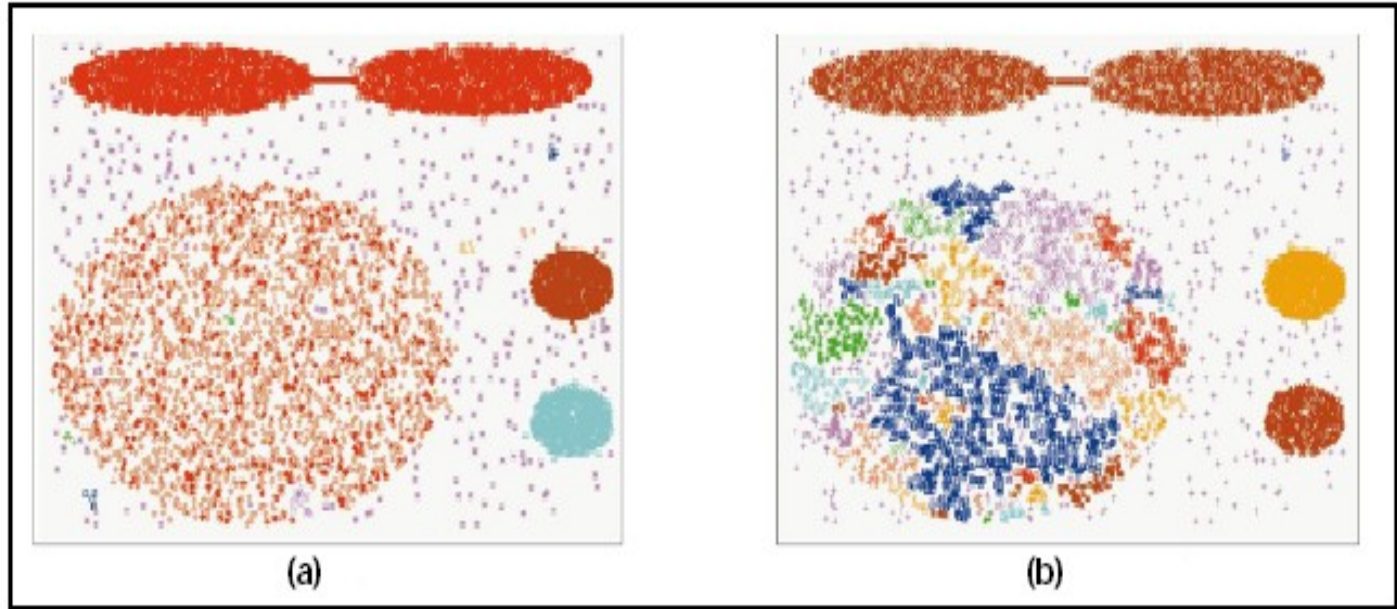
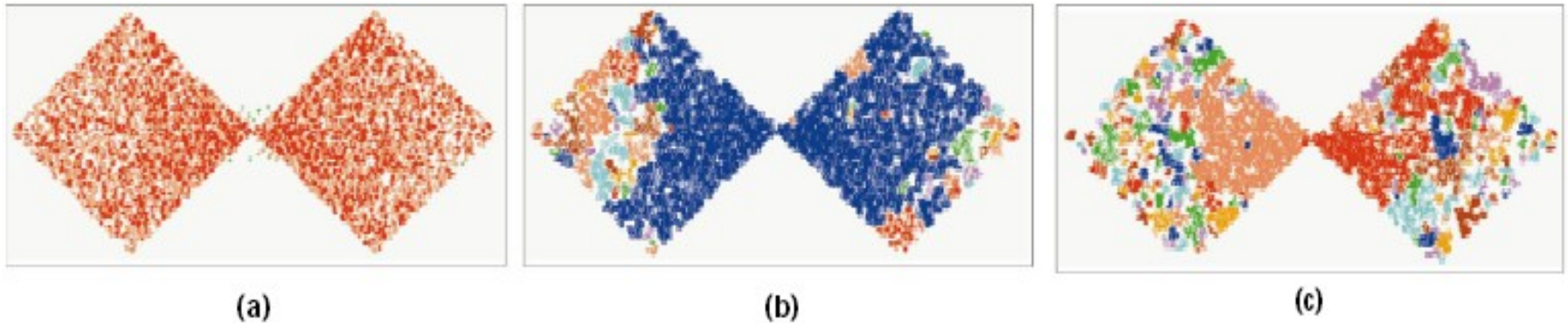


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



DBSCAN online Demo:

<http://webdocs.cs.ualberta.ca/~yaling/Cluster/Applet/Code/Cluster.html>

OPTICS: A Cluster-Ordering Method (1999)

- OPTICS: Ordering Points To Identify the Clustering Structure
 - Ankerst, Breunig, Kriegel, and Sander (SIGMOD'99)
 - Produces a special order of the database wrt its density-based clustering structure
 - This cluster-ordering contains info equiv to the density-based clusterings corresponding to a broad range of parameter settings
 - Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
 - Can be represented graphically or using visualization techniques

DENCLUE: Using Statistical Density Functions

- DENSity-based CLUstEring by Hinneburg & Keim (KDD'98)
- Using statistical density functions:

$$f_{Gaussian}(x, y) = e^{-\frac{d(x, y)^2}{2s^2}}$$

influence of y on x

$$f_{Gaussian}^D(x) = \sum_{i=1}^N e^{-\frac{d(x, x_i)^2}{2\sigma^2}}$$

total influence on x

$$\nabla f_{Gaussian}^D(x, x_i) = \sum_{i=1}^N (x_i - x) \cdot e^{-\frac{d(x, x_i)^2}{2\sigma^2}}$$

gradient of x in the direction of x_i

- Major features

- Uses gaussian *kernel density approximation*

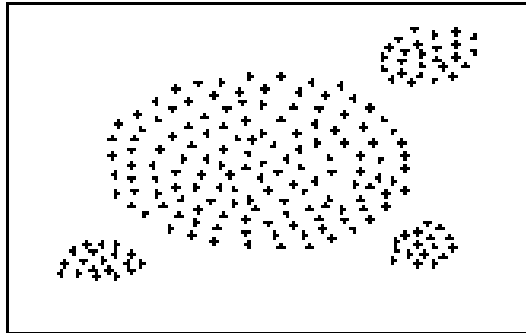
$$f(x) = \frac{1}{nS} \sum_{i=1}^n K\left(\frac{x - x_i}{S}\right)$$

- Clusters can be determined mathematically by identifying density attractors (local maxima of the overall density function)
- Center defined clusters: assign to each density attractor the points density attracted to it (pick each point and follow the gradient)

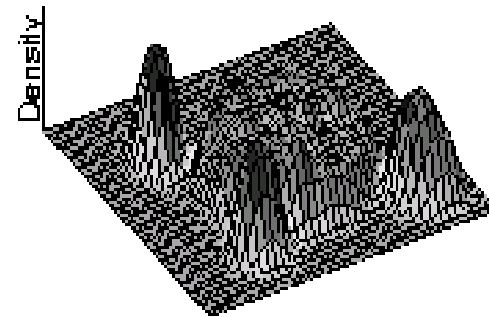
Denclue: Technical Essence

- Arbitrary shaped cluster: merge density attractors that are connected through paths of high density ($>$ threshold)
- Solid mathematical foundation
- Good for data sets with large amounts of noise
- Allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets
- Significant faster than existing algorithm (e.g., DBSCAN)
- But needs a large number of parameters

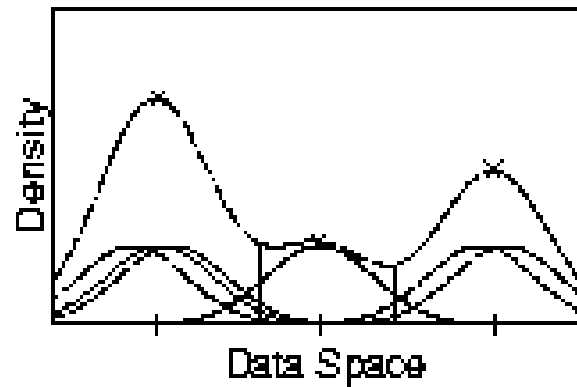
Density Attractor



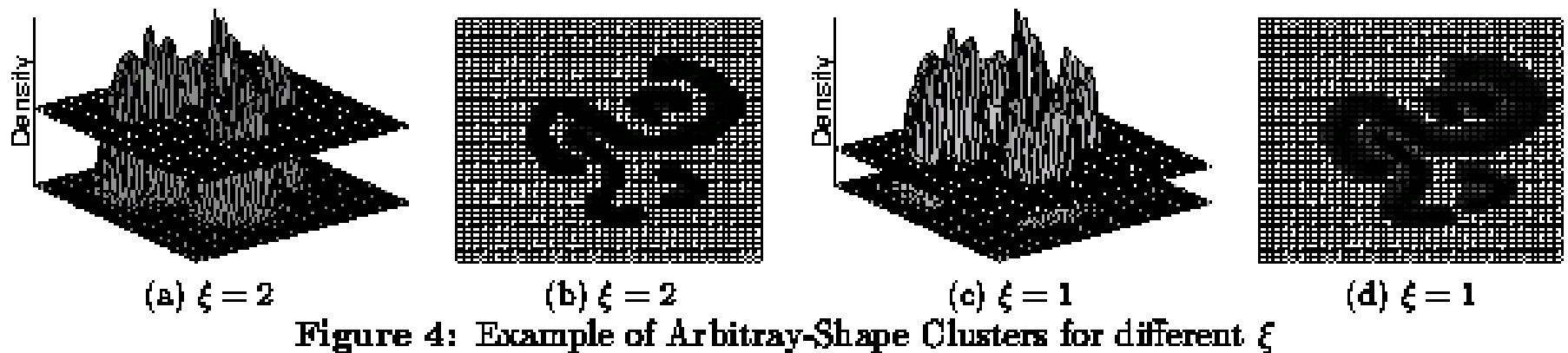
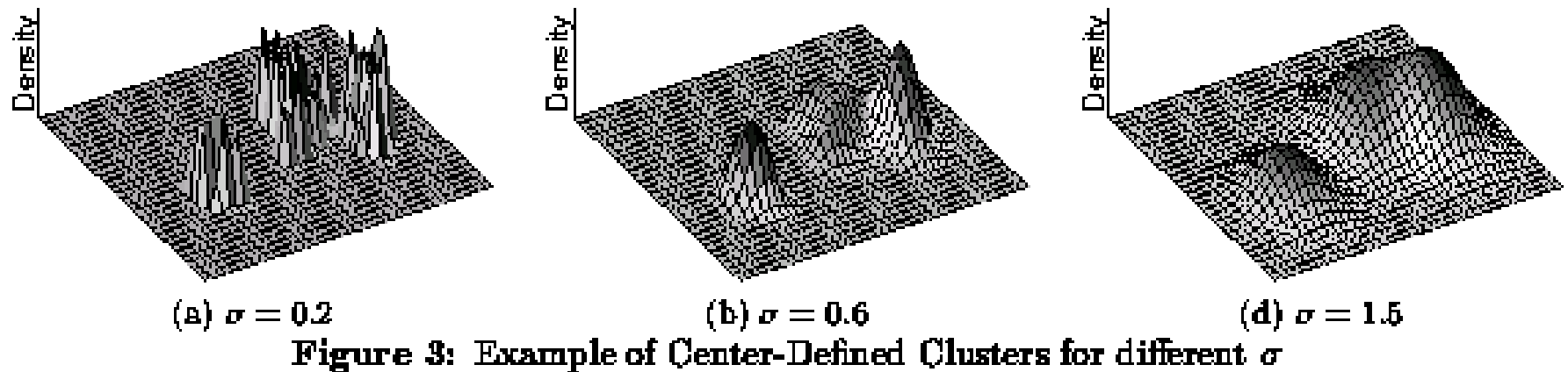
(a) Data Set




(c) Gaussian



Center-Defined and Arbitrary



Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods 
- Evaluation of Clustering
- Summary

Grid-Based Clustering Method

- Using multi-resolution grid data structure
- Several interesting methods
 - **STING** (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
 - **WaveCluster** by Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
 - A multi-resolution clustering approach using wavelet method
 - **CLIQUE**: Agrawal, et al. (SIGMOD'98)
 - Both grid-based and subspace clustering

STING: A Statistical Information Grid Approach

- Wang, Yang and Muntz (VLDB'97)
- The spatial area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution

The STING Clustering Method

- Each cell at a high level is partitioned into a number of smaller cells in the next lower level
- Statistical info of each cell is calculated and stored beforehand and is used to answer queries
- Parameters of higher level cells can be easily calculated from parameters of lower level cell
 - *count, mean, s, min, max*
 - type of distribution—*normal, uniform, etc.*
- Use a top-down approach to answer spatial data queries
- Start from a pre-selected layer—typically with a small number of cells
- For each cell in the current level compute the confidence interval

STING Algorithm and Its Analysis

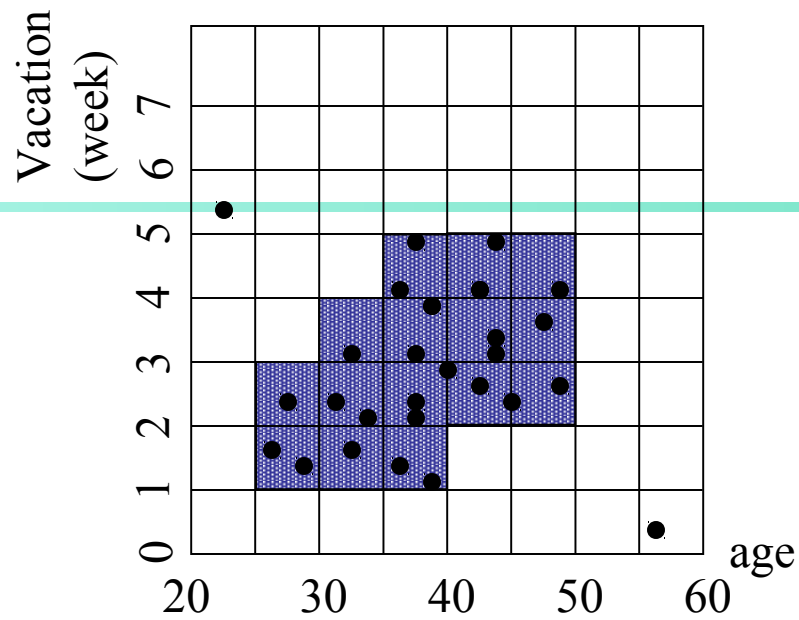
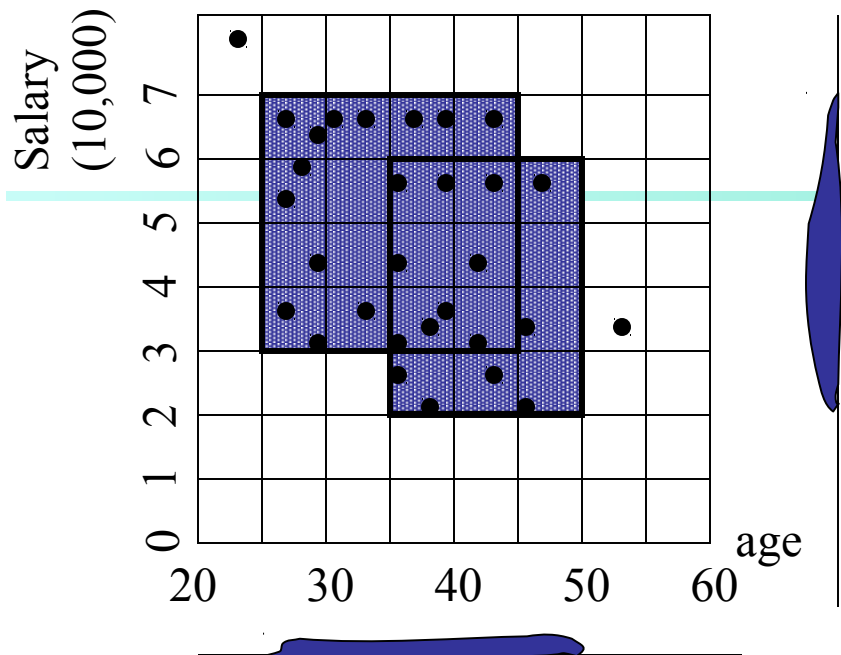
- Remove the irrelevant cells from further consideration
- When finish examining the current layer, proceed to the next lower level
- Repeat this process until the bottom layer is reached
- Advantages:
 - Query-independent, easy to parallelize, incremental update
 - $O(K)$, where K is the number of grid cells at the lowest level
- Disadvantages:
 - All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected

CLIQUE (Clustering In QUES)

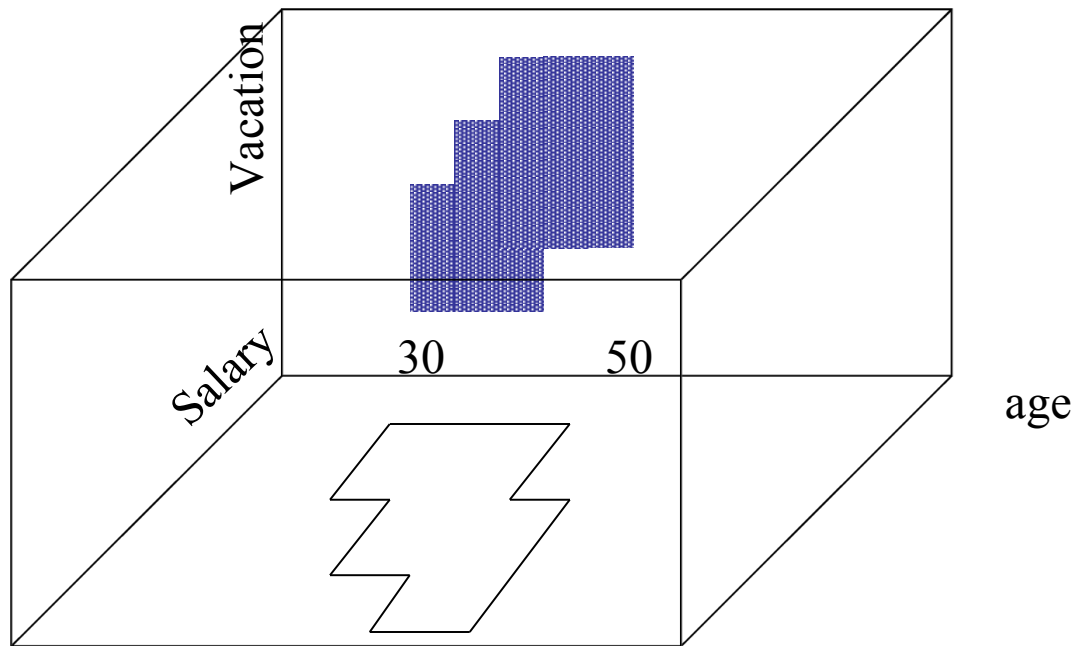
- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98)
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
 - It partitions each dimension into the same number of equal length interval
 - It partitions an m-dimensional data space into non-overlapping rectangular units
 - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
 - A cluster is a maximal set of connected dense units within a subspace

CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
- Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters
 - Determine dense units in all subspaces of interests
 - Determine connected dense units in all subspaces of interests.
- Generate minimal description for the clusters
 - Determine maximal regions that cover a cluster of connected dense units for each cluster
 - Determination of minimal cover for each cluster



$\tau = 3$



Strength and Weakness of *CLIQUE*

- Strength
 - *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
 - *insensitive* to the order of records in input and does not presume some canonical data distribution
 - scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases
- Weakness
 - The accuracy of the clustering result may be degraded at the expense of simplicity of the method